

Desenvolvimento do PhorFun: uma proposta de *website* para divulgação de eventos

Ana Beatriz F. de Jesus, André C. da Silva

¹Grupo de Pesquisa Mobilidade e Novas Tecnologias de Interação
Curso de Análise e Desenvolvimento de Sistemas
Instituto Federal de Ciência e Tecnologia de São Paulo (IFSP)
Avenida Thereza Ana Cecon Breda, N.º 1896,
Vila São Pedro – 13183-250 – Hortolândia – SP – Brasil

jesus.ana@aluno.ifsp.edu.br, andre.constantino@ifsp.edu.br

Abstract. *This article addresses the development of a website prototype for promoting events called Phorfun. After the COVID-19 pandemic and all the negative effects effects on tourism and the events sector, these sectors are growing again, this time and this time with the help of the media to promote themselves. The defined methodology is based on studies of articles with similar themes and objectives, as well as a basic study of all the tools used during development. In the end, the result is a complete application planning and a prototype of the application with some functionalities implemented.*

Resumo. *Este artigo aborda o desenvolvimento de um protótipo de website para divulgação de eventos chamado Phorfun. Após a pandemia da COVID-19 e todos os efeitos negativos sentidos no turismo e no setor de eventos, esses setores voltam a crescer e dessa vez contando com o auxílio das mídias para se divulgarem. A metodologia usada foi baseada em estudos de artigos de temas e objetivos semelhantes, além do estudo básico de todas as ferramentas utilizadas durante o desenvolvimento. Por fim o resultado é um planejamento completo da aplicação e um protótipo da aplicação com algumas funcionalidades implementadas.*

1. Introdução

Após o início da pandemia do COVID-19 no Brasil em 2020, houve uma queda de R\$ 270 bilhões somando o ano de 2020 e 2021 com uma taxa de desemprego muito elevada na área de eventos, alcançando 3 milhões de pessoas [Kaoru 2021].

Ainda nesse período de pandemia o setor de eventos tenta crescer novamente, dessa vez utilizando mais e mais a *Internet* como aliada na divulgação. Durante a pandemia foi comum o uso de *livestreams* para "shows" realizados de forma virtual. [Dino 2023] apresenta uma pesquisa realizada com profissionais estimando que o número de eventos presenciais retornariam aos níveis pré pandemia em um ou dois anos. Já em 2023, de acordo com a [Sampaio 2023], foi realizado um levantamento da Associação Brasileira dos Promotores de Eventos (Abrape) baseado em dados coletados pelo IBGE e do Ministério do Trabalho, demonstrando que o setor registrou, de janeiro a junho, 42,3% contratações a mais em comparação com o mesmo período em 2022.

Este projeto está dividido em seções, sendo essa primeira a introdução com Objetivo e Justificativa, onde será explicado qual a meta a ser alcançada ao desenvolver este projeto e a motivação do mesmo, respectivamente. Logo após, será apresentado o Referencial Teórico, com explicações de termos técnicos, metodologias e ferramentas utilizadas no desenvolvimento deste projeto; então há os Requisitos do Sistema, onde serão apresentados requisitos funcionais e não funcionais referentes à proposta do projeto, os Trabalhos Correlatos que serviram de inspiração e guia para o desenvolvimento, plataformas e artigos que auxiliaram no planejamento do projeto. Em seguida a Metodologia, onde é explicado com maior detalhamento como as ferramentas foram utilizadas dentro desse projeto, o Desenvolvimento que relata todos os passos desde a escrita da primeira linha deste artigo até o desenvolvimento prático final da aplicação e, por fim, a Conclusão, o resultado alcançado tanto da aplicação quanto do conhecimento articulado durante o desenvolvimento do mesmo.

1.1. Objetivo

A proposta é o desenvolvimento do protótipo de um *website* em que se possa divulgar diversos tipos de eventos, organizá-los por categoria e poder compartilhar estes com o círculo social do usuário sem a necessidade de compartilhar *links* por aplicativos externos. Esses eventos serão exclusivamente divulgados sem a venda direta de ingressos ou qualquer outra forma de garantir acesso aos próprios. Será possível divulgar eventos pequenos como festas em família e possuir confirmação apenas para convidados escolhidos pelo usuário que criou o evento.

1.2. Justificativa

Esse *website* para divulgação de eventos poderá alcançar diversos públicos para que possa criar interesse no usuário por eventos diferentes. Assim, será um ganho, tanto para o usuário com novas experiências quanto para os eventos por conta do público maior e diversificado. Além disso, esse *website* irá ter o controle de confirmação para eventos menores, como festas de aniversário; festas em finais de semana, eliminando a necessidade do usuário de buscar confirmação diretamente com cada convidado.

A maior importância na divulgação de um evento aberto ao público é que a propaganda chegue aos interessados levando em conta seu tema, ambiente, custo e distância de trajeto. Esse tipo de propaganda é conhecida por “propaganda direcionada”, onde uma coleta de dados dos rastros do usuário pela *Internet*. Serviços como *AdSense*, do Google, que mapeiam rastros online como cliques e compras, vão direcionando propagandas de acordo com o interesse do usuário. Por fim essas informações são compartilhadas com outras empresas criando uma rede de propagandas direcionadas em diversas aplicações e *sites online* [Lavado 2019].

O foco no público alvo é citado por [Canton 1997] “Nenhum negócio será duradouro se [...] não levar em conta a necessidade de adaptar o seu serviço às transições, expectativas e necessidades de seus cliente”.

2. Referencial Teórico

Nessa seção serão abordados o *framework* e o método utilizados para divisão de tarefas do projeto e organização de tempo como Scrum e Kanban. Também serão descritas ferramentas utilizadas durante o desenvolvimento do projeto como Bootstrap, Nodemon e

Express. Essas ferramentas foram utilizadas no desenvolvimento prático da aplicação, sendo o Nodemon um facilitador ao fazer alterações em linguagem JavaScript usando Node.js como *framework*. Já o Bootstrap foi utilizado para o desenvolvimento em nível de código das interfaces de usuário.

2.1. Scrum

De acordo com [Schwaber and Sutherland 2020] "*Scrum* é um *framework* leve que ajuda pessoas, times e organizações a gerar valor por meio de soluções adaptativas para problemas complexos". Esse framework (Figura 1) conta com eventos e papéis a serem exercidos, e possui pilares empíricos, sendo: transparência, inspeção e adaptação. Esses pilares possuem uma relação de dependência entre si, onde a inspeção precisa da transparência e a adaptação depende da inspeção. A inspeção só pode ser feita quando há honestidade e a transparência em todas as etapas do processo, onde serão realmente relatados todas as soluções e problemas encontrados. A adaptação só pode ser feita depois de analisada a inspeção e identificado o que há para melhorar, mudar ou descartar na maneira a se realizar as tarefas do processo.

O *Scrum Team* é composto por um *Scrum Master*, um *Product Owner* e *Developers*, sendo normalmente uma equipe de 10 pessoas ou menos, todos focados na Meta do Produto. Os seguintes papéis são definidos:

1. *Developers*: papel desempenhado por pessoas responsáveis por transformar o *Product Backlog* (descrição das funcionalidades do produto) em um produto funcional.
2. *Product Owner*: papel desempenhado por quem irá maximizar o valor do produto e fazer alterações no *Product Backlog* se necessário.
3. *Scrum Master*: papel desempenhado pelo responsável por entender e aplicar o *framework Scrum*. É responsável por fazer o *framework* funcionar de fato, auxiliando o time a melhorar práticas e liderando a equipe.

O *Scrum* conta com os seguintes eventos:

1. *Sprint*: evento que faz o *Scrum* acontecer, costuma ter duração fixa, comumente de 15 dias, mas pode ser de um mês ou menos e inicia logo após a *Sprint* anterior acabar. É onde os *Developers* desenvolvem o produto de acordo com o *Product Backlog*.
2. *Sprint Planning*: evento onde é definido o que será feito na *Sprint*, o *Product Owner* irá discutir o *Product Backlog* nesse momento com a equipe e será planejado o caminho a seguir para chegar na Meta do Produto. Pode ser guiada pelas seguintes perguntas: Por que a *Sprint* é valiosa? O que pode ser feito durante a *Sprint*? De que maneira o trabalho definido pode ser feito?
3. *Daily Scrum*: evento onde geralmente o *Scrum Master* irá guiar a equipe. É uma reunião que deve ser feita diariamente em horário e local fixo, com duração máxima de 15 minutos. As principais perguntas para os participantes são: O que foi feito ontem? O que será feito hoje? Há algum impedimento? É importante que seja uma reunião rápida, direta e sucinta, onde os *Developers* se sintam confortáveis para compartilhar soluções e dificuldades que estão passando.
4. *Sprint Review*: esse evento é responsável por analisar o resultado obtido ao final da *Sprint*. O *Product Backlog* pode sofrer alterações nesse momento e, por fim, é analisado o progresso em relação à Meta do Produto.

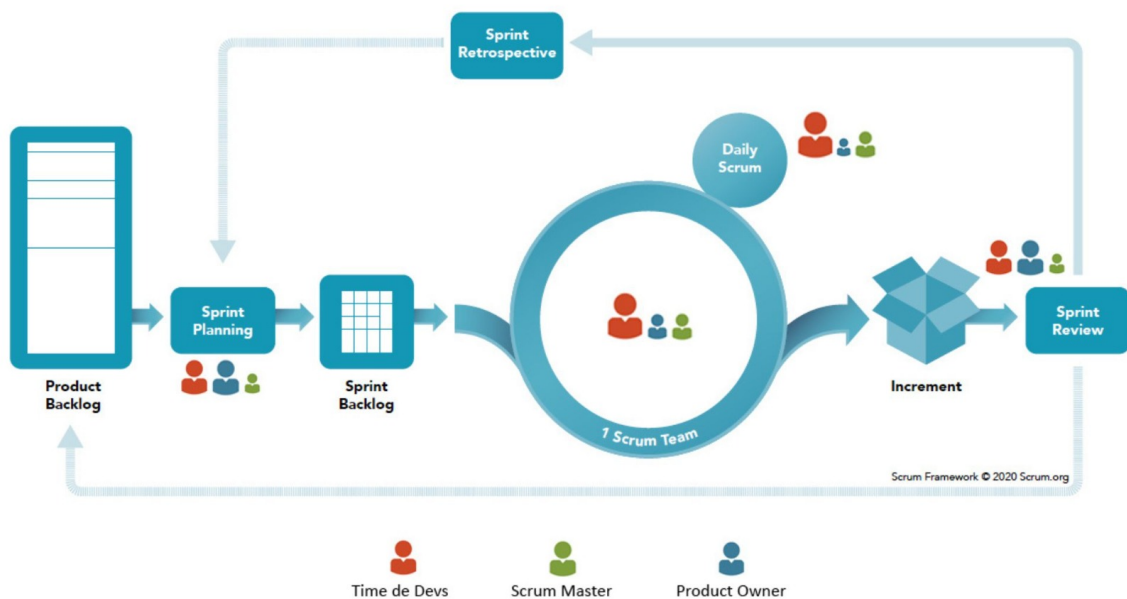


Figura 1. Modelo Scrum. Fonte: [Schwaber and Sutherland 2020]

5. *Sprint Retrospective*: último evento do Scrum. A partir das informações coletadas na *Sprint Review*, o time irá analisar melhorias que podem ser feitas para a próxima *Sprint*. Esse evento é mais focado no processo que no resultado, onde os problemas serão discutidos com atenção, tanto os que foram solucionados quanto os que não foram.

2.2. Kanban

De acordo com [Kanbanize 2023] e [Ferreira 2020] Kanban é um método para gerir o trabalho que, no começo do século 21 foi notado como possível ferramenta útil para melhorar a entrega de produtos e serviços na indústria do *software*. Kanban, traduzindo literalmente do japonês, significa “sinalização” ou “cartão”. Desenvolvido e posteriormente aplicado pela Toyota em 2007 em sua versão mais simples. O Kanban mais simples possui “Pedido”, “Em Progresso” e “Concluído” como colunas. Posteriormente as tarefas são adicionadas nas colunas correspondentes ao seu estado atual. Essa organização destaca as informações quase em tempo real, mostrando o estado geral e as etapas do processo.

As práticas do Kanban, ainda de acordo com [Ferreira 2020] são:

- Visualizar o Fluxo de Trabalho;
- Limite o Trabalho em Progresso (WIP);
- Gerencie o Fluxo;
- Deixe as Políticas de Processo Explícitas;
- Implemente Ciclos de *Feedbacks*,
- Melhore Colaborativamente.

No geral, essas práticas seguem a linha de colocar à mostra o trabalho que está sendo feito, assim será mais fácil notar onde pode ser melhorado, onde está dando certo e onde está falhando. O foco deve ser sempre no processo e não em quem está realizando a

atividade. O limite de trabalho em andamento é necessário no Kanban, pois multitarefas podem gerar desperdício e ineficiência. Também é preciso restringir a quantidade de tarefas em uma etapa do processo pois o excesso pode encobrir algum problema de processo. Todos os envolvidos no processo devem o compreender e estar familiarizados com ele. Os *feedbacks* permitem que haja análise e planejamento de mudanças se necessário. E por fim deve se cultivar uma cultura organizacional onde todos entendam que são importantes no processo, que suas dificuldades são importantes e assim moldar o trabalho para que atenda ao cliente e seja eficiente no desenvolvimento [Ferreira 2020].

2.3. React

Desenvolvido pelo Facebook, React é uma biblioteca muitas vezes confundida com *framework* sendo até mesma comparada com *frameworks* como Angular, em pesquisas como [Awari 2022] foi apontado como um dos *frameworks* mais populares em 2022. Por ser muito flexível, ter um ótimo desempenho e grande demanda de mercado, é uma escolha comum de profissionais ao iniciar a carreira, pois transforma o desenvolvimento web o tornando muito mais eficiente. O React se baseia em componentes, utiliza um conceito denominado Virtual DOM (*Document Object Model*), que seria uma representação da estrutura de uma página web, ou seja, o React cria uma representação da estrutura DOM e compara com DOM base, só alterando o necessário, atualizando de forma eficiente, além disso a biblioteca utiliza JSX, que seria uma extensão da sintaxe do JavaScript, permitindo escrever HTML dentro de seu código, por fim, utiliza de state e props que são fundamentais para o desenvolvimento utilizando a biblioteca, sendo state um dado mutável de estado do componente e props utilizado para passar dados entre componentes de “pai” para “filho” [de Camargos et al. 2019].

2.4. Node.js

Node.js é um *framework back-end* que foi desenvolvido em 2009 por Ryan Dahl e outras 14 pessoas. O objetivo era desenvolver uma tecnologia com arquitetura não bloqueante, pois outros sistemas possuem essa características que suspende o processamento enquanto aguardam uma entrada ou saída de dados no servidor. Então a arquitetura planejada permite processos simultâneos sem cadeia de ação. É importante ressaltar que o Node.js tem uma estrutura orientada a eventos, além da forma de entrada e saída que permite que seja leve e eficiente. Node.js também permite uma escalabilidade, pois as características citadas são essenciais para aplicações em tempo real [Barsoti and Gibertoni 2020].

2.5. Ferramentas de Desenvolvimento

Nesta seção serão descritas as ferramentas utilizadas durante o desenvolvimento da aplicação, desde o *design* até ferramentas de *frameworks* para melhor escrita do código.

Nodemon é uma ferramenta de apoio usada durante o desenvolvimento de um projeto Node.js. Tem como função reiniciar aplicativos a cada alteração no projeto garantindo certa agilidade ao codificar [Vargas et al. 2022].

Express é um *framework Web* que facilita o desenvolvimento de APIs no Node.js. Com esse *framework* se torna mais fácil organizar partes da aplicação usando funções como o *middleware* por exemplo [Barsoti and Gibertoni 2020].

Bootstrap é uma ferramenta desenvolvida em 2010 com objetivo de auxiliar no *front-end* do Twitter. É de código aberto mantida por uma pequena equipe de desenvolvedores. Seu diferencial, maior além de seus códigos prontos para estilização em HTML e CSS é a responsividade de todas as opções de componentes. É muito utilizada quando o objetivo é estilizar de modo fácil e rápido aplicações inteiras, sendo sua desvantagem o fato de que se mantém um padrão, mas permite a edição dentro do CSS para manter a aplicação com uma interface "única". Uma curiosidade é que qualquer usuário pode contribuir na biblioteca enviando uma *issue* ou *pull request*, a qual será avaliada pela equipe e pode ser implementada em uma nova versão [Bootstrap 2023].

Figma é uma ferramenta com foco em *design* e prototipagem colaborativa. Sua interface é intuitiva mas possui recursos avançados, sendo ótima para projetos escolares até profissionais. É uma ferramenta que possui armazenagem *online*, então é possível trabalhar em conjunto com uma equipe mesmo de forma remota. Sendo uma ferramenta inclusive de uso profissional, é comum o uso de *plugins* e até responsividade automática dentro da mesma [Awari 2023].

2.6. Requisitos do Sistema

De acordo com [Institute 2017]: "os requisitos não funcionais complementam os requisitos funcionais e descrevem as condições ou qualidades ambientais requeridas para que o produto seja eficaz.". Já os requisitos funcionais são os que descrevem os serviços ou funções que o sistema deve ter, como o sistema deve lidar com cada tipo de entrada.

Com base nos conceitos de [Sommerville 2018] serão descritos os requisitos não funcionais que deverão ser aplicados no sistema a ser desenvolvido. Alguns desses requisitos, como escalabilidade, poderão ficar pendentes para trabalhos futuros. O sistema será desenvolvido para um público amplo no entanto em pequena escala nesse primeiro momento.

- **Manutenibilidade:** Atributo crítico a longo prazo. O software deve ser desenvolvido de forma que possa receber alterações e evoluções com o tempo.
- **Aceitabilidade:** Deve ser aceitável para o usuário alvo, sendo inteligível, compatível com outros sistemas que o usuário utiliza e útil.
- **Desempenho:** Relativo ao tempo de resposta do sistema.
- **Confiabilidade:** Manter a taxa de falhas baixa e aceitável. Pode ser medida através de uma média de tempo até ocorrer a falha e probabilidade do sistema ficar indisponível.
- **Segurança:** Segurança da informação que o sistema possui, ou seja, impedir vazamento de dados ou alterações indevidas.
- **Robustez:** Relativo ao tempo para reinicialização após falha do sistema. Também pode ser medido e checado de acordo com a porcentagem de eventos falhos e probabilidade de corromper dados.
- **Usabilidade:** De acordo com [Winckler and Pimenta 2002] é relativo à qualidade da interação usuário x interface. Afirma também que pode ser associada à facilidade de aprendizado, facilidade de memorizar a forma de realizar tarefas, satisfação subjetiva do usuário, etc.
- **Escalabilidade:** É definida por [de Carvalho Neto 2017] como a habilidade do sistema em se adaptar e suportar grande volume de carga sem falhas na performance.

3. Trabalhos Correlatos

3.1. Linguagem JavaScript como Alternativa para o Desenvolvimento de Aplicações Multiplataformas

O artigo busca explicar as vantagens e melhores formas de se usar o JavaScript para desenvolvimento de aplicações multiplataforma. A metodologia utilizada foi análise de livros, estudo sobre artigos relacionados, entre outros. É apresentado o uso da linguagem JavaScript como alternativa para dinamizar as páginas *Web*, que antes de sua criação, eram estáticas [da Silva Cruz et al. 2018].

A Google [2018] caracterizou o V8 (um motor JavaScript de código aberto - *open source*, basicamente o mesmo é responsável por executar os códigos JavaScript dentro do navegador Google Chrome, ou seja, ela irá ler os códigos escritos e compilar para *byte code*). Todo navegador possui um motor JavaScript. O V8 é referente ao Google Chrome como mecanismo de código aberto escrito em C++ de execução de JavaScript de alto desempenho. A partir disso, é considerada sua implementação no próprio navegador (Google Chrome) e no Node.js podendo ser executado em Windows, MacOS e Linux. De acordo com Flanagan[2011], Node é um rápido interpretador JavaScript baseado em C++ com ligações às APIs de baixo nível do Unix para trabalhar com processos, arquivos, etc.

Node.js é um ambiente de execução multi-plataforma em JavaScript que permite desenvolvedores produzirem aplicações *server-side* para rede usando JavaScript como linguagem. Além disso, Node.js se difere de outras linguagens de *server-side*, pois é orientada a eventos, é *mono-thread*(utiliza um processo, o principal). Também é abordado desenvolvimento *desktop* com a biblioteca Electron, a estrutura Vuidoa que serve para criar aplicativos de *desktop* nativos baseados no Vue.js e outros [da Silva Cruz et al. 2018].

Os benefícios de utilizar JavaScript são principalmente dois. A escalabilidade, que se define pela disponibilidade de serviços requisitados a proporções crescentes de novos usuários; a performance, desempenho obtido considerando a escalabilidade e recursos consumidos.

Este artigo influenciou na escolha da linguagem para o desenvolvimento do projeto. Até mesmo alguns elementos de tela poderão ser desenvolvidos em React, uma biblioteca do JavaScript como foco em interface.

3.2. Sympla

Sympla é uma plataforma de eventos ampla que atende mais de 3.500 cidades brasileiras, sendo referência na área. A plataforma permite que seja criada uma conta e o usuário pode tanto participar de eventos e fazer compras diretamente no aplicativo quanto divulgar seu próprio evento de forma gratuita. A Figura 2 mostra a tela inicial da aplicação *Web* do Sympla, apresentando em destaque o menu. Enquanto isso a Figura 3 apresenta algumas telas do Sympla em sua versão *mobile*.

Iniciada em 2012 pelos irmão Rodrigo e Marcelo Cartacho e David Tomasella, a plataforma tem o objetivo de conectar pessoas por meio da tecnologia. Teve seu primeiro escritório em Belo Horizonte. Em 2016, entrou para o grupo Movable e adquiriu seu concorrente da época, Eventick. Em 2020 estende seu serviço para transmissão ao vivo e videoconferência via Zoom, Sympla Streaming e Sympla Play e se torna a plataforma de eventos mais famosa do Brasil[Cartacho et al. 2023].

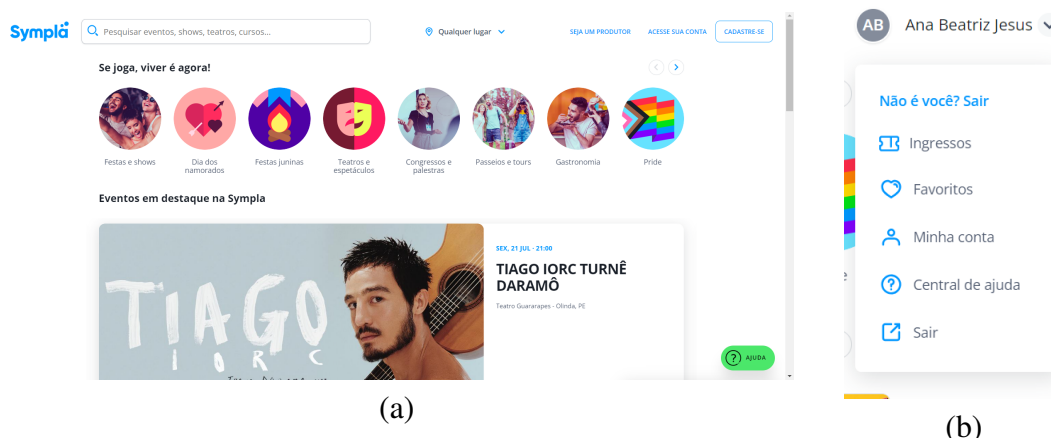


Figura 2. Recortes de telas do Sympla Web (a) tela inicial e (b) menu

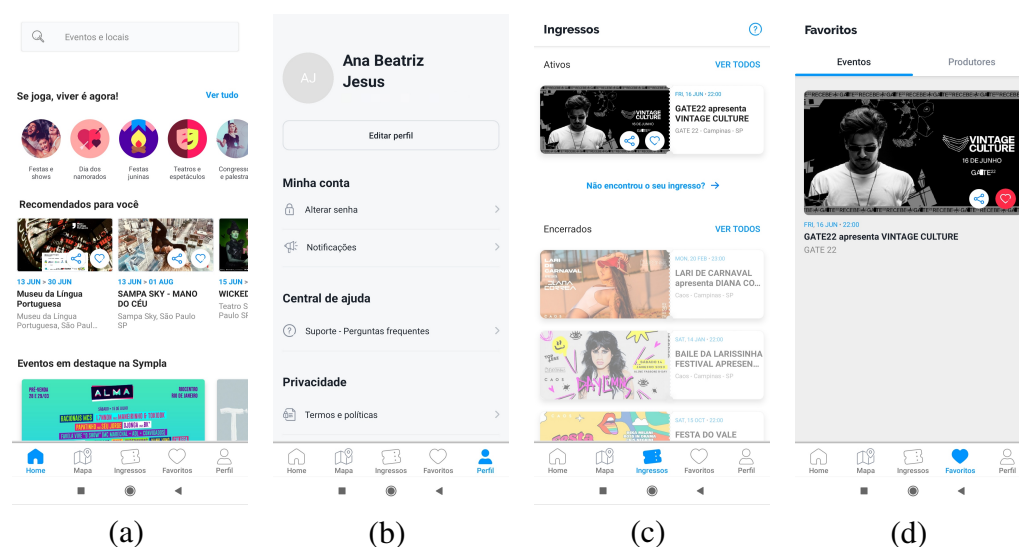


Figura 3. Telas do aplicativo Sympla para Mobile (a) tela inicial, (b) menu, (c) lista de ingressos e (d) favoritos

Sympyla é executado tanto em *desktop*, *website* e aplicativos Android e IOS. Para utilizar o usuário pode fazer seu cadastro, visualizar eventos e favoritá-los (conforme Figura 4), bem como utilizar a localização e checar eventos próximos e também é possível a criação grátis de um novo evento, mas para eventos que envolvam compras de ingresso há uma taxa cobrada pela plataforma de R\$2,50 [Cartacho et al. 2023].

3.3. Ingresso

Outra grande plataforma de divulgação e venda de ingressos de eventos que buscou na tecnologia uma forma de conectar as pessoas e suas experiências e memórias. Fundada em 2015 com 4 valores principais baseados em ícones e não conceitos: aliança, bússola, cálice e escudo. Possui centrais em Brasília, Goiás, Rio Grande do Sul e São Paulo.[Ingresso 2023]

Ingresso tem seu foco em vender ingressos. Para utilizar a plataforma que também está disponível para Android e IOS (Figura 6), além da Web (Figura 5), o usuário deve se cadastrar. Após isso, poderá buscar eventos por estado brasileiro ou pelo nome do evento,

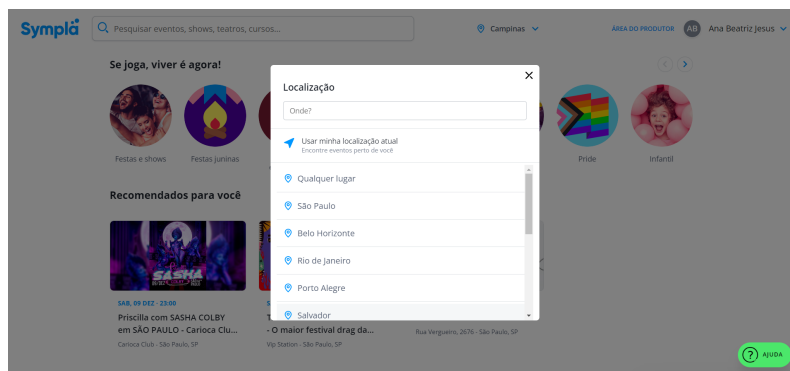


Figura 4. Busca de Eventos por localização Symplic Web

visualizar eventos separados por categoria na tela inicial, adquirir um ingresso dentro da plataforma e visualizar esse ingresso na "carteira" dentro da aplicação.

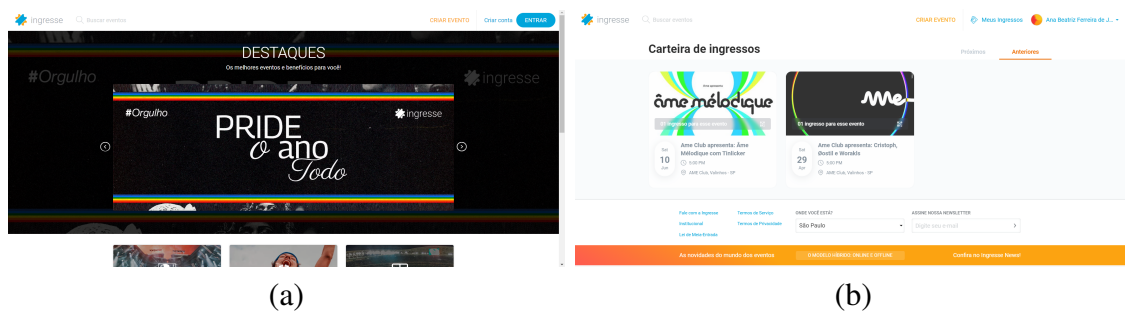


Figura 5. Telas do Ingresso Web (a) inicial e (b) carteira de ingressos

3.4. Byma

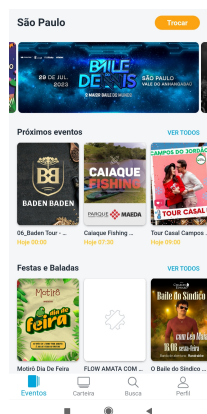
Byma é uma plataforma de divulgação e venda de ingressos com foco em grandes eventos. Essa plataforma, diferente das outras citadas, precisa que o usuário se torne um parceiro da Byma para poder cadastrar e divulgar seu evento. Há um formulário que precisa ser preenchido e será avaliado pela empresa.[Byma 2023]

É uma plataforma mais simples e com menos funções, também disponível para Android e IOS além da Web. Após se cadastrar, é possível ao usuário visualizar os eventos na tela inicial e buscar por algum específico. Se adquirido um ingresso, esse estará visível na aba Ingressos tendo uma separação entre eventos que já ocorreram e eventos que irão ocorrer (vide Figuras 7a, 7b e 7c).

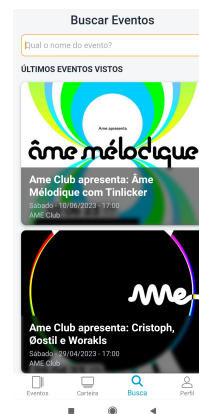
Por fim foi realizada uma análise (Tabela 1) das funções de cada plataforma estudada e das funções planejadas para a aplicação Phorfun.

4. Metodologia

A metodologia utilizada foi baseada no *Scrum*, apresentado na Subseção 2.1. O projeto foi dividido em *Sprints* de 15 dias onde o orientador assume o papel de *Product Owner* e irá maximizar o valor do Produto e o aluno assume o papel de *Developer*. As reuniões são feitas ao fim de cada *Sprint* e semelhante ao *Sprint Review* e será checado o que foi feito nessa *Sprint*, quais as dificuldades e o que será feito na próxima *Sprint*.

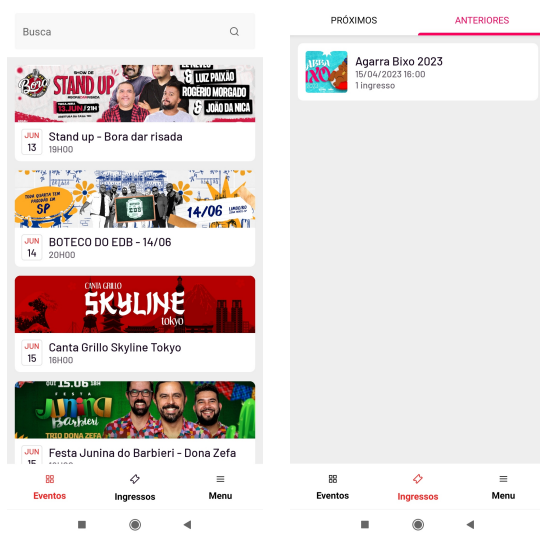


(a)



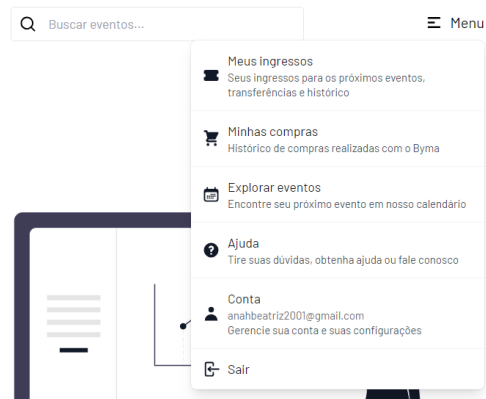
(b)

Figura 6. Tela do Ingresso Mobile (a) inicial e (b) carteira de ingressos



(a)

(b)



(c)

Figura 7. Telas do Byma (a) inicial para Web, (b) ingressos para Mobile e (c) menu para Web

Para o desenvolvimento será utilizada principalmente a linguagem JavaScript em conjunto com HTML e CSS. Para uso efetivo da linguagem JavaScript serão utilizadas extensões como Express e Nodemon, já para desenvolvimento do CSS e HTML, será utilizado elementos do Bootstrap.

Aplicações semelhantes como Ingresso, Sympla e Byma foram analisadas para extrair quais dados são necessários para o cadastro do usuário, quais as funções que um usuário comum tem acesso, a distribuição de informações na interface, como eventos no *feed*, ao entrar no evento, onde se localiza o endereço, a hora que ocorrerá, a descrição do evento, etc.

5. Desenvolvimento

Durante a primeira *Sprint* foram analisadas plataformas com objetivos e funcionalidades semelhantes para consolidar a ideia do produto. Foi sugerido a criação de uma tabela

Tabela 1. Tabela comparativa entre as aplicações PhorFun, Sympla, Ingresse e Byma.

	PhorFun	Sympla	Ingresse	Byma
Cadastro Usuário	✓	✓	✓	✓
Visualizar Eventos	✓	✓	✓	✓
Visualizar Ingresso / Evento	✓	✓	✓	✓
Mapa de Eventos Próximos	✗	✓	✗	✗
Eventos Favoritados	✗	✓	✗	✗
Relação entre Usuários	✓	✗	✗	✗
Criação de Eventos Grátis	✓	✓	✓	✗
Busca de Evento	✓	✓	✓	✓
Busca por Localização	✗	✓	✓	✓
Buscar Categoria	✓	✓	✓	✗
Evento Privado/ Pequeno	✓	✓	✗	✗

comparativa onde seriam relatados os resultados dos testes em plataformas semelhantes e comparassem com a aplicação proposta. Analisando a Tabela 1, observa-se que a aplicação que possui mais funcionalidades dentre as de interesse é o Sympla, seguido pelo Ingresse. Destacamos que o Sympla possibilita a gerência de eventos privados ou pequenos, enquanto que o Ingresse e o Byma não. Esta é uma característica que desejamos para o PhorFun.

Durante a segunda *Sprint* foi desenvolvido o diagrama de casos de uso (Figura 8) e um modelo entidade-relacionamento (Figura 9), seguindo as funcionalidades identificadas dadas na primeira reunião.

Ao ator "Convidado" foram identificadas as funcionalidades recusar ou confirmar convite e visualizar eventos que o usuário tenha recebido convite. Ao ator "Cliente" foram identificadas as funcionalidades de procurar eventos e visualizar eventos gerais. Ao ator "Organizador" foi identificada a funcionalidade de requisitar ao Administrador alterações no evento criado. E por fim foi criado o ator "Administrador" para que houvesse o con-

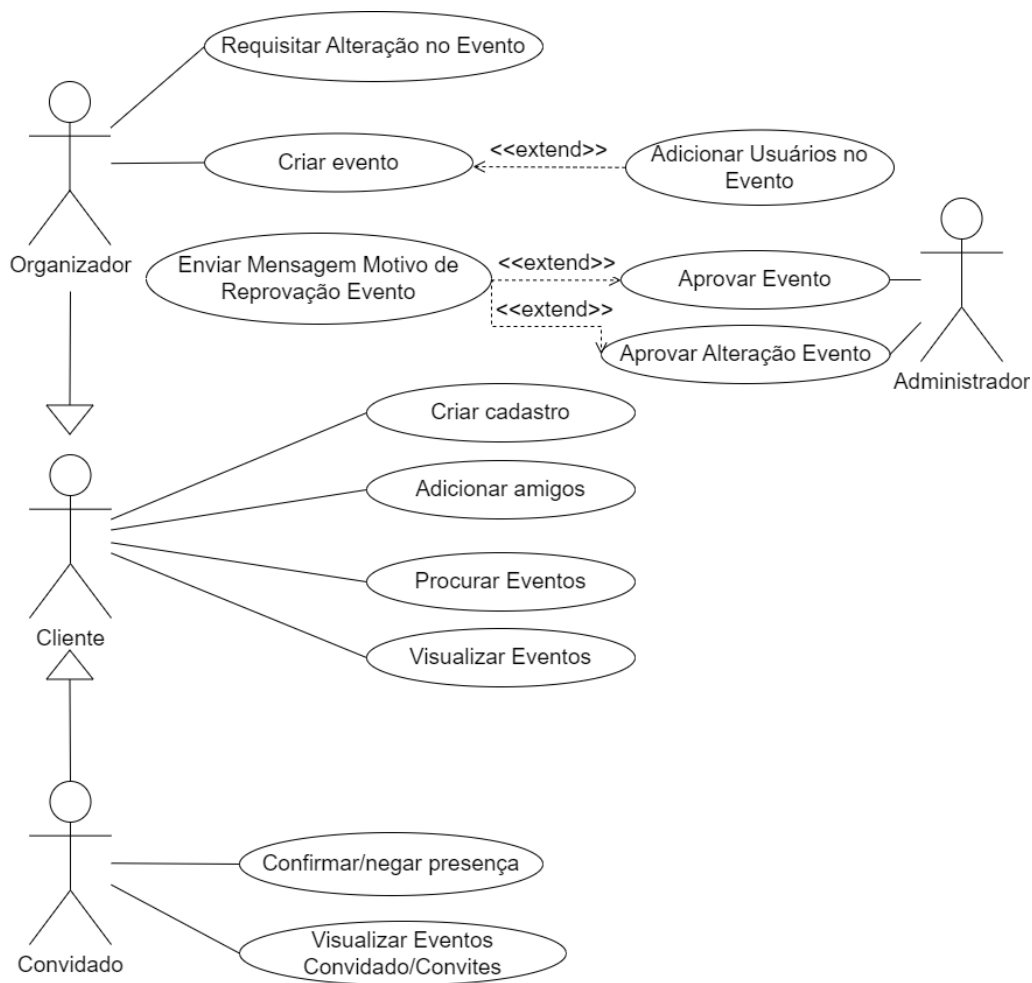


Figura 8. Diagrama de Casos de Uso para o PhorFun

trole do tipo de evento a ser divulgado na plataforma e controle de alterações no evento, ou seja, um Organizador não poderia criar um evento com características para ser aprovado e depois alterar de acordo com sua vontade. Pensando nessa situação, o ator "Administrador" teve em si identificadas as funcionalidades de aprovar ou não um evento, enviando um *feedback* ao Organizador para que este entendesse o motivo de seu evento não ser aprovado, e a função de aprovar alteração no evento, tendo também a possibilidade de dar um retorno do motivo da não aprovação ao Organizador.

Durante a terceira *Sprint* foram desenvolvidos alguns protótipos de telas para o PhorFun (conforme as Figuras 11 e 12).

Durante a quarta *Sprint* foram pensadas novas tarefas para o quadro Kanban para controle das atividades a serem realizadas durante todo o fim do projeto e durante cada *Sprint*, com colunas "A Fazer", "Sprint Atual", "Fazendo", "Esperando Aprovação" e "Concluído". Esse quadro está programado usando *labels* em cada tarefa para reconhecimento fácil do estado da tarefa. Além disso há um controle na coluna "Fazendo", onde só será permitida a adição de três atividades, como um limite para controle do que está realmente sendo feito. Além de tarefas com *checklist* interno exclusivamente para controle de desenvolvimento das diversas telas.

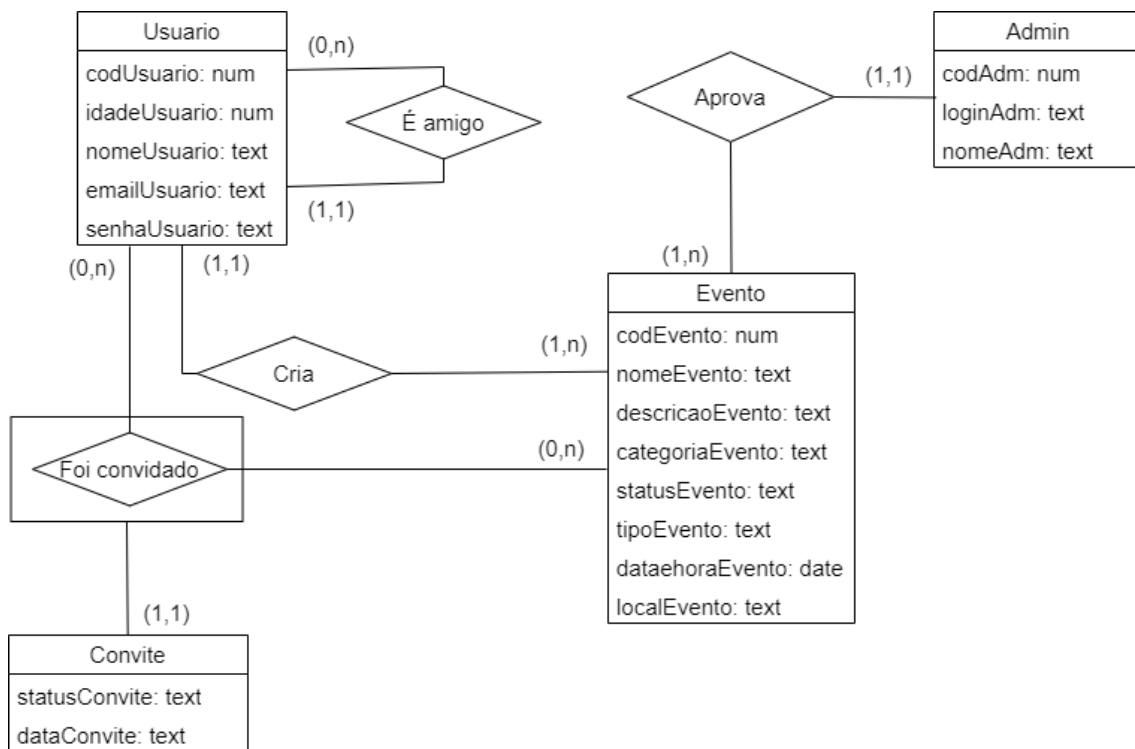


Figura 9. Diagrama entidade-relacionamento para o PhorFun

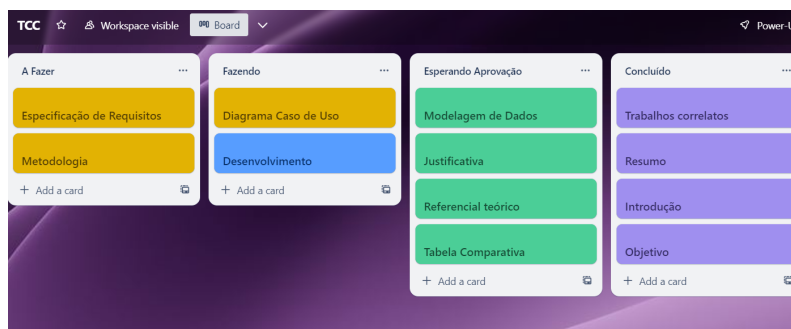


Figura 10. Quadro Kanban do Projeto

Durante a quinta *Sprint* foram desenvolvidas alguns protótipos de telas como *Login*, Cadastro, Inicial de Usuário, Criação de Eventos e Visualização de Eventos Criados para serem usadas como guia ao desenvolver a interface a nível de código.

Durante a sexta *Sprint* foi criado o banco de dados usando o Sistema Gerenciador de Banco de Dados PostgreSQL. O diagrama inicial necessitou de alterações na tabela de amizade do usuário (vide Figura 13). A mudança foi a criação de uma tabela amizade, onde será registrado um convite de amizade, com um estado (aceito ou recusado ou não respondido). Essa mudança foi necessária para facilitar no retorno de amizades de cada usuário.

Durante a sétima *Sprint* foi criado o projeto React da aplicação e implementado uma parte da interface de usuário. Houveram dificuldades na integração do Bootstrap no projeto, pois o Bootstrap comum não poderia ser utilizado dentro da linguagem JavaS-



Figura 11. Telas do Phorfun para Web (a) inicial e (b) cadastro de novo usuário

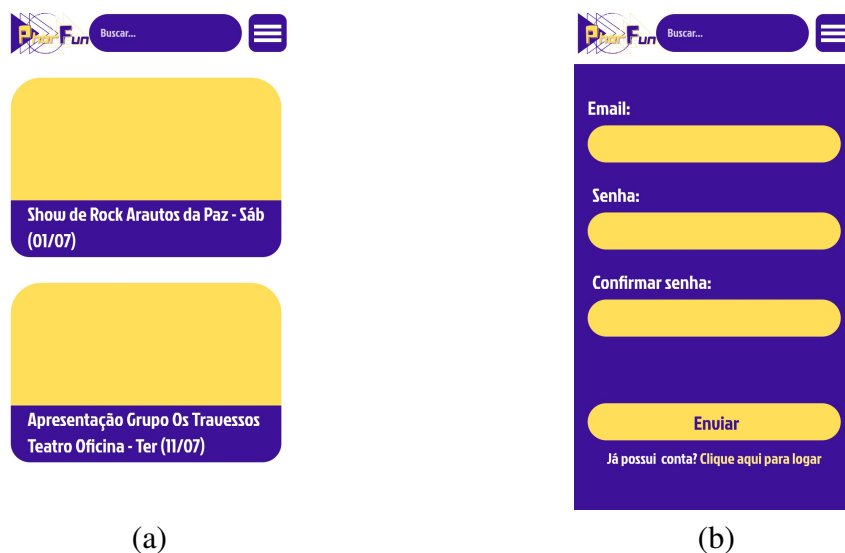


Figura 12. Telas do PhorFun Mobile (a) inicial e (b) cadastro de usuário

cript, então foi pesquisado o Bootstrap React e este foi integrado perfeitamente, sendo necessário estudar essa biblioteca por ser uma versão diferente. Para a integração do Bootstrap React foi enviado o comando "npm install react-bootstrap bootstrap" via terminal e utilizado o seguinte trecho de código no arquivo js na Figura 14.

Na linha 1 da Figura 14 há a importação do Bootstrap React, ou seja, é necessária a internet para que seus componentes sejam utilizados, é utilizado para aplicar os componentes dentro do trecho HTML e por fim, nas linhas 2 a 10 são importados cada elemento que será utilizado de forma individual.

Durante a oitava Sprint foram iniciadas as codificações das funções para cadastro e login, sendo já criado o projeto em Node.js. Foram instaladas extensões como Nodemon e Express para facilitar o desenvolvimento. A interface foi adiada para o final, sendo consideradas mais importantes as funcionalidades no back-end. O Nodemon foi utilizado para não se fazer necessário a utilização do comando "node server.js" a cada alteração no código, onde foi colocado o trecho "dev": "nodemon src/server.js" dentro da sessão scripts no arquivo package.json e ao enviar no terminal o comando "node run dev" a

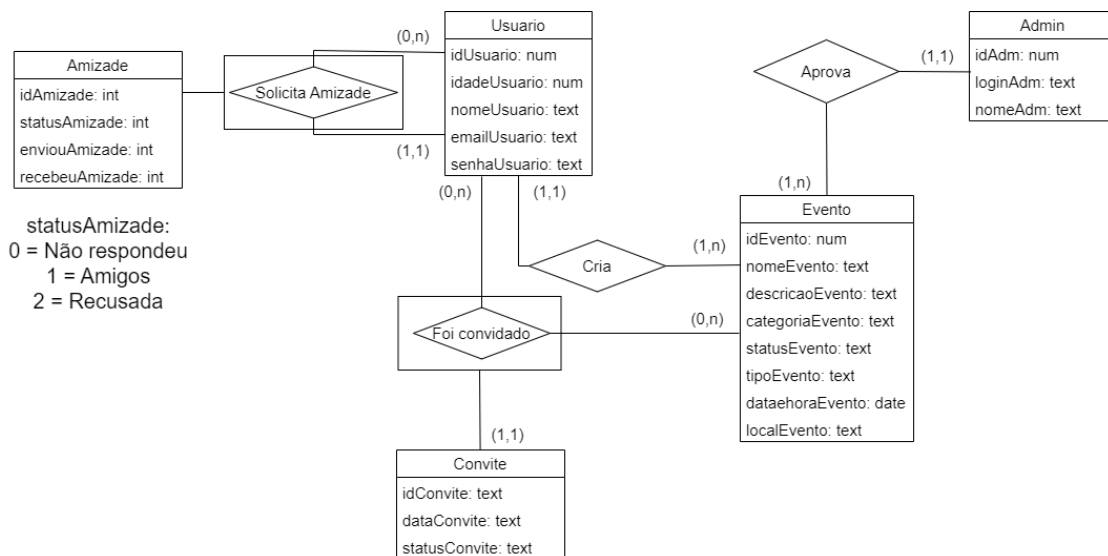


Figura 13. Diagrama entidade-relacionamento do PhorFun atualizado

```

1. import 'bootstrap/dist/css/bootstrap.min.css';
2. import {
3.   Collapse,
4.   Navbar,
5.   NavbarToggler,
6.   NavbarBrand,
7.   Nav,
8.   NavItem,
9.   NavLink
10. } from 'reactstrap';

```

Figura 14. Código de importação da biblioteca Bootstrap React

página se atualizava a cada alteração salva. A Figura 15 Um exemplo de código da função de registro, onde é checado se o *e-mail* do usuário é pré-existente na base de dados e caso seja, é retornada a mensagem de erro.

Na linha 1 é definida a rota que será analisada (ou seja, chamada via POST com a URI /register). Nas linhas 2 e 3 são obtidos os dados de *e-mail* e login passados como parâmetro na requisição. A linha 4 cria a consulta SQL que irá buscar por um registro com aquele e-mail na tabela Usuario. A linha 5 executa a consulta SQL e, se houver erro, envia o erro ao cliente (linhas 6 e 7), e caso a consulta execute com sucesso, os dados referente ao usuário com o e-mail fornecido são retornados para o cliente (linha 9).

Durante a nona *Sprint* foi continuada a codificação das funções para cadastro e *login*. Durante o desenvolvimento o banco de dados foi migrado para MySQL devido a maior facilidade. A Figura 16 apresenta o código de conexão.

Na linha 1 é feito a importação da biblioteca MySQL2 que foi previamente instalada utilizando o comando “npm install mysql2” no terminal. Na linha 2 é utilizada uma função da biblioteca MySQL2, e as linhas seguintes se referem à porta do banco de dados

```

1. app.post("/register", (req, res) => {
2.   const email = req.body.email;
3.   const senha = req.body.senha;
4.   db.query("SELECT * FROM usuario WHERE emailUsuario = ?", [email],
5.   (err, result)=> {
6.     if (err) {
7.       res.send(err);
8.     }
9.     res.send(result);
10.  });
11. });

```

Figura 15. Código em Node.Js para verificar se um determinado e-mail já possui registro no sistema.

```

1. const mysql = require('mysql2');
2. const db = mysql.createPool({
3.   host: 'localhost',
4.   port: 3306,
5.   user: 'root',
6.   password: '',
7.   database: 'phorfun',
8. });

```

Figura 16. Código em Node.Js para criação de conexão com o SGBD MySQL

e informações do usuário como nome e senha(censurada), e por fim na linha 7 o nome da base de dados.

6. Conclusões

Este trabalho teve como objetivo desenvolver um protótipo de aplicação para a divulgação de eventos chamado PhorFun. Foram identificadas funcionalidades e comparações com softwares similares, cujos requisitos funcionais foram expostos por meio de diagrama de casos de uso. Outros artefatos também foram criados, como o diagrama entidade-relacionamento para o banco de dados e o design de todas as interfaces na plataforma Figma. No entanto, a primeira versão da aplicação resultante desse trabalho consta com duas funcionalidades implementadas tanto no *back-end* (Node.js) como no *front-end* (React), duas interfaces que utilizam o Bootstrap React, garantindo responsividade em diversas dimensões de tela, código padronizado para facilitar desenvolvimento futuro e garantir manutenibilidade do sistema.

Durante o desenvolvimento houveram muitas escolhas repensadas, principalmente na ordem das tarefas definidas, como o objetivo de fazer o *front-end* antes do *back-end*, o que demandou muito tempo que poderia ser empregado em funcionalidades e trabalhado a interface somente depois, tendo apenas o básico dos campos para demonstrar o funcionamento da aplicação.

Outras funcionalidades ficam pendentes para trabalhos futuros, onde a aplicação será desenvolvida com maior tempo de estudo da linguagem utilizada, e a análise no

decorrer desse desenvolvimento os requisitos não funcionais como aceitabilidade, confiabilidade, robustez e escalabilidade.

Ao decorrer deste projeto, foram utilizados muitos conhecimentos adquiridos no curso de Análise e Desenvolvimento de Sistemas. Para a escolha de metodologia que melhor se aplicasse, foi usado o conhecimento de *Scrum* aprendido na disciplina de Metodologias Ágeis, assim como o quadro Kanban, que serviu como uma ótima escolha para organização de tarefas. O diagrama de casos de uso e o levantamento de requisitos foram desenvolvidos a partir do aprendizado em Análise Orientada a Objetos e Engenharia de Software. Para o desenvolvimento a nível de código e a modelagem de banco de dados, Banco de Dados I e II foram disciplinas muito enriquecedoras bem como Algoritmos e Programação e Desenvolvimento Web.

Referências

- Awari (2022). As melhores frameworks frontend de 2022 para impulsionar sua carreira tech. <https://awari.com.br/as-melhores-frameworks-frontend-de-2022-para-impulsionar-sua-carreira-tech/>. [Online; Acessado em 17 de dezembro de 2023].
- Awari (2023). Domine o figma: um guia completo para aprendizado e sucesso. <https://awari.com.br/domine-o-figma-um-guia-completo-para-a-prendizado-e-sucesso/>. [Online; Acessado em 10 de outubro de 2023].
- Barsoti, N. and Gibertoni, D. (2020). Impacto que o Sequelize traz para o desenvolvimento de uma API construída em Node.JS com Express.JS. *Revista Interface Tecnológica*, 17.
- Bootstrap (2023). Bootstrap. <https://getbootstrap.com/docs/5.3/about/overview/>. [Online; Acessado em 21 de outubro de 2023].
- Byma (2023). Byma. <https://byma.com.br/>. [Online; Acessado em 12 de junho de 2023].
- Canton, M. (1997). Evento: da proposta ao planejamento. *Revista Turismo em Análise*, 8(1):18–30.
- Cartacho, R., Cartacho, M., and Tomasella, D. (2023). Sympla. <https://www.sympla.com.br/>. [Online; Acessado em 12 de junho de 2023].
- da Silva Cruz, V., Petrucelli, E. E., and Sotto, E. C. S. (2018). A linguagem javascript como alternativa para o desenvolvimento de aplicações multiplataforma. *Revista Interface Tecnológica*, 15(2):39–49.
- de Camargos, J. G. C., Coelho, J. F., Aramuni, J. P., and Villela, H. F. (2019). Uma análise comparativa entre os frameworks javascript angular e react. *Computação Sociedade*, 1.
- de Carvalho Neto, L. R. (2017). Escalabilidade em aplicações web: estudo de caso em um sistema ruby on rails. *Universidade Federal do Rio Grande do Norte*.
- Dino (2023). Ano começa promissor para o setor de eventos. <https://valor.globo.com/patrocinado/dino/noticia/2023/02/14/ano-comeca-promissor-para-o-setor-de-eventos.ghtml>. [Online; Acessado em 5 de maio de 2023].

- Ferreira, M. B. (2020). *Métodos ágeis e melhorias de processos*, volume 1. Contentus.
- Ingresse (2023). Ingresse. <https://sobre.ingresse.com/sobre-nos>. [Online; Acessado em 12 de junho de 2023].
- Institute, P. M. (2017). *Um Guia do Conhecimento em Gerenciamento de Projetos*, volume 6. Project Management Institute.
- Kanbanize (2023). Guia do usuário kanbanize. <https://knowledgebase.kanbanize.com/hc/en-us/articles/6002160259986-Guia-do-Usu%C3%A1rio-Kanbanize#:~:text=Kanbanize%20%C3%A9%20o%20um%20software,de%20maneira%20r%C3%A1pida%20e%20eficiente>. [Online; Acessado em 5 de maio de 2023].
- Kaoru, T. (2021). Setor de eventos perde R\$ 270 bilhões durante a pandemia. <https://www.cnnbrasil.com.br/economia/setor-de-eventos-perde-r-270-bilhoes-durante-a-pandemia/>. [Online; Acessado em 26 de março de 2023].
- Lavado, T. (2019). As empresas de tecnologia te escutam para vender publicidade? É mais provável que elas nem precisem. <https://g1.globo.com/economia/tecnologia/noticia/2019/10/15/saiba-se-as-empresas-de-tecnologia-escutam-suas-conversas.ghtml>. [Online; Acessado em 5 de dezembro de 2023].
- Sampaio, A. (2023). Setor de eventos foi um dos principais geradores de emprego no Brasil no primeiro semestre, diz associação. <https://www.cnnbrasil.com.br/economia/setor-de-eventos-foi-um-dos-principais-geradores-de-emprego-no-brasil-no-primeiro-semester-diz-abrape/>. [Online; Acessado em 5 de dezembro de 2023].
- Schwaber, K. and Sutherland, J. (2020). *O guia do scrum*.
- Sommerville, I. (2018). *Engenharia de Software*, volume 10. Editora Pearson.
- Vargas, R. M., Rego, P. D. E., and Viegas, S. C. (2022). Projeto de sistema para integração da API JIRA com web service. *REFAQI - Revista de Gestão de Educação e Tecnologia*, 13.
- Winckler, M. and Pimenta, M. S. (2002). Avaliação de usabilidade de sites web. *Escola de Informática da SBC Sul*, 1:85–137.

Documento Digitalizado Público

Anexo I - artigo - TCC

Assunto: Anexo I - artigo - TCC
Assinado por: Andre Constantino
Tipo do Documento: Relatório
Situação: Finalizado
Nível de Acesso: Público
Tipo do Conferência: Documento Digital

Documento assinado eletronicamente por:

- **Andre Constantino da Silva, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 20/03/2024 13:39:07.

Este documento foi armazenado no SUAP em 20/03/2024. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifsp.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 1618163

Código de Autenticação: 6f920e47a5

